AMENDMENT TO THE CLAIMS:

The following listing of claims will replace all prior versions of claims in the application:

Claims 1 – 20 (Canceled)

21. (New) A method for on demand functional verification of a software model (40) of an application specific integrated circuit (ASIC), wherein said software model is written in a low–level programming language and separately causes a model of the circuit to be generated and debugging of functional verification tests to be applied to the model of the circuit for constituting a verification platform, comprising a transmission mode and a verification mode:

- creating in the transmission mode an autonomous circuit emulator (1), obtained by replacing the software model (40) which is in a low level programming language physically describing the circuit under design to be validated with a high level language abstract description generating response data structures in accordance with a functional specification (20) of the project as a function of the stimuli received, this mode being called the "transmission mode."

- integrating the software model (40) in a verification mode into a verification platform, and connecting previously validated autonomous circuit emulator (1), in parallel, to interfaces of the software model (40) of the circuit, and to an environment emulator (11, 21, 22); and

- utilizing the verification platform as a reference for the validation of response data transmitted by the software model (40) of the circuit.

22. (New) A method according to claim 21, wherein a user:

- generates, using a data processing system, the autonomous circuit emulator (1) which provides a simulation configuration corresponding to the software model (40) of the ASIC using the functional specification (20),
- writes, from the functional specification (20), and stores in a test platform (21, 22, 23) for integrated circuit models, a program (51) for testing the software model (40) of the ASIC, comprising input stimuli sequences to be provided to the software model (40) of the ASIC, which

the autonomous simulation configuration (1), based on the functional specification (20), corresponds to output stimuli sequences,

- links together, and activates, the autonomous simulation configuration (1) and the test platform (21, 22, 23), and
- observes the output stimuli of the HDL-type model (40) of the ASIC in order to functionally validate the system constituted by the software model of the ASIC circuit and the validation test program (210), and thus validates the software model (40) in comparison to the functional specification (20) .

23. (New) A method according to claim 21, wherein the autonomous configuration (1) communicates with the user to control the activation of previously created and stored models of input stimuli sequences defined in a high-level programming language, and controls the activation of associated programs (90) for the progressive validation of test sequences determined from the models.

24. (New) A method according to claim 21, wherein a user writes and provides the functional specification (20) in a low-level programming language, specifying functional models of circuits.

25. (New) A method according to claim 22, wherein the user writes and provides the functional specification (20) in a low-level programming language, specifying functional models of circuits.

26. (New) A method according to claim 23, wherein a user writes and provides the functional specification (20) in a low-level programming language, specifying functional models of circuits.

27. (New) A method according to claim 21, wherein a user provides the functional specification (20), in the form of a program in a low level programming language of functional models of circuits, and a program in a high level programming language of functional models of circuits, and the user controls the

autonomous simulation configuration (1) so as to perform a co-simulation by synchronizing the execution of the two specification programs.

28. (New) A method according to claim 27 wherein the low level language is an HDL type and the high level language C^{++}.

29. (New) A method according to claim 21 wherein the verification platform verifies that the responses of the software model of the ASIC are within response time ranges specified in the functional specification (20).

30. (New) A method according to claim 22 wherein the verification platform verifies that the responses of the software model of the ASIC are within response time ranges specified in the functional specification (20).

31. (New) A method according to claim 23 wherein the verification platform verifies that the responses of the software model of the ASIC are within response time ranges specified in the functional specification (20).

32. (New) A verification platform for on demand verification of a software model of an integrated circuit (ASIC), characterized in that it comprises data processing means that allow a client to select test models producing input stimuli for the ASIC, said data processing means being constructed and arranged to read functional specification elements (20) of the ASIC and comprising programs (90) that form an emulator and generate a functional validation test program (51) constituted by output stimuli, from the input stimuli and the functional specification elements (20).

33. (New) A verification platform according to claim 32, comprising a library of functional models of circuit blocks for ASICs and means for selecting models through a definition file of the integrated circuit configuration, in order to create a model corresponding to the functional specification of the ASIC that is integrated into the definition of its environment.

34. (New) A verification platform according to claim 32, further including, in a link connecting the platform to a client, two serial programming language adaptation circuits (11, 12), that transform commands in a high level programming language (C^{++}), used by the client, into commands in a low level programming language used by the ASIC model, and respectively, to the commands in the low level programming language back into commands in a high level programming language.

35. (New) A verification platform according to claim 32, characterized in that the platform includes means (90, 10) for executing operations at the same time as the simulation, and upon detection of an error interrupting operations at the very moment the error appears.

36. (New) A verification platform according to claim 32, characterized in that the functional specification elements (20) are constituted by a truth table corresponding to the functions of the various functional circuit elements of the ASIC software model (40), and a propagation delay to be respected between each input and each output.

37. (New) A verification platform according to claim 32, characterized in that the functional specification elements (20) are constituted by a behavior table corresponding to the functions of the various functional circuit elements of the ASIC software model (40), and a propagation delay to be respected between each input and each output.

38. (New) A verification platform according to claim 32, further including a cache memory (962) for storing the blocks used by nodes according to node addresses, and means for managing, for an address used by one or more nodes, a presence vector with one presence indicator per node.

39. (New) A verification platform according to claim 38 characterized in that the programs (90) are object-oriented and the emulator is structured as a set of classes for managing a collection of execution hypotheses for a transaction in a memory block of the software model, and for managing transactions that are concurrently colliding using the same memory block.

40. (New) A verification platform according to claim 38, characterized in that algorithms of the programs (90) of the emulator perform the following functions: generating predictions, eliminating predictions, readjusting incorrect predictions, reducing the number of valid hypotheses, and terminating collisions

41 (New). A verification platform according to claim 40, characterized in that the emulator of the circuit generates predictions without having to obtain additional information on the internal operation of the circuit under design.

42. (New) A verification platform according to claim 38, characterized in that the platform is used as an emulator of a router circuit, a circuit with cache or a router circuit with cache.

43. (New) A verification platform according to claim 38 for testing a software model of an integrated circuit (ASIC) on demand characterized in that the platform comprises an ASIC emulator (1) for controlling a comparator (23) that receives values generated by a software model of the ASIC circuit tested, upon reception of stimuli sent by at least one stimuli generating circuit (21) storing a test program, an interface (11) for translating the stimuli from an advanced language into a low level language corresponding to that of the software model, and means for validating the verification in case of the detection of a collision by the comparator (23).

44. (New) A verification platform according to claim 38, further comprising means for selecting the response to stimuli that depend on the composition of the circuits tested, said means for selecting being constituted by a model generated by

means for selecting functional models from a library, which associates with each of the models the responses to a given stimulus, the model corresponding to the composition of the circuit to be tested.

45. (New) A verification platform according to claim 44, further including means (7) for storing responses selected so as to create a test model (70) to be applied to the

46. (New) A verification platform according to claim 32, characterized in that each transaction is constituted, at the level of each interface, by a request packet and one or more associated response packets, wherein the values of the parameters and/or the transmission time constraints of the packets can be forced from the functional test program executed by the emulator of the environment, which appropriately translates all of said parameters during the transmission of the packets to the terminals of the software model of the design.

47. (New) A verification platform according to claim 43 characterized in that each transaction is constituted, at the level of each interface, by a request packet and one or more associated response packets, wherein the values of the parameters and/or the transmission time constraints of the packets can be forced from the functional test program executed by the emulator of the environment, which appropriately translates all of said parameters during the transmission of the packets to the terminals of the software model of the design.